CLAIMS

Having thus described the invention, what is claimed is:

1. A system that allows analysis of software running in a tamper- resistant environment, the system comprising:

a processor which monitors activities and creates a log entry for at least one of a set of selected activities;

an encryption system which encrypts the log entry for the at least one selected activity;

a log of a relatively-fixed size which stores the encrypted log entries; and

random data in the log file when it is originally created and which is replaced by log entries so that the size of the log containing log entries appears to be a substantially-constant size; and

a pointer which identified the next storage location for the next log entry so that the last log entry can be determined and the next log entry can be positioned in a location in the log after the previous log entry.

2. A system including the elements of Claim 1 wherein the system includes a transmission system for sending the log file, upon command, to a secure processing location away from the system in which the log file was created.

3. A system including the elements of Claim 1 wherein the system includes a system for wrapping around and filling the log from the beginning when the log file has been filled, allowing the log file to remain at a substantially-constant size even after the log file has been filled with data and a new entry is received.

4. A system including the elements of Claim 1 wherein the system includes a mechanism for obscuring the activity for which a log entry is created.

5. A system including the elements of Claim 4 wherein the mechanism for obscuring the activity for which a log entry is created includes a printing function for writing into the log file.

6. A system including the elements of Claim 2 wherein the system includes a mechanism for receiving an indication from the user that transmission is desired and transmits the file in response to that indication.

7. A system including the elements of Claim 1 wherein the system further includes a mechanism for receiving an input from a user that initiates logging of log entries into the log.

8. A system including the elements of Claim 1 wherein the system further includes an initializing mechanism for determining when logging is to begin and initiating logging of log entries only in response to that initializing mechanism.

9. A system including the elements of Claim 1 wherein the system uses a public key to provide the log entries and a private key corresponding to the public key is used to decrypt the entries at a secure location.

10. A method of analyzing the operation of software in a tamper-resistant environment comprising the steps of:

generating a log file full of random data;

turning on logging and establishing a pointer for the location of the next logged event;

monitoring the operation of software within the tamper-resistant environment and generating messages in response to operation of the software within the tamper-resistant environment;

logging an event relating to a generated message by replacing a random data with an encrypted record of an event;

moving the pointer when a log entry has been made to the next available log position;

wrapping the pointer to the top of the file when the log is full of log entries; and

sending the log to a secure location where it may be decrypted and analyzed; and

analyzing the decrypted data and providing information on the operation of the software in the tamper-resistant environment.

11. A method including the steps of Claim 10 wherein the step of turning on logging includes the steps of receiving an user input that logging is desired and initiating the logging in response thereto.

12. A method including the steps of Claim 10 wherein the step of logging an event further includes the steps of determining whether the event is to be logged, and if so, determining when to log the event to obscure what is being logged.

13. A method including the steps of Claim 10 wherein the step of logging an even further includes the steps of determining the next location for logging, replacing the existing data in the location with the data from the event, and updating the pointer to provide the location of the next logged event.

14. A method including the steps of Claim 10 and further including the step of receiving a command from a user that indicates that sending the log file to a remote location is desired and transmitting the log file in response thereto.

15. A service which operates to analyze the operation of software in a remote protected processing environment, the service including:

receiving from the remote protected processing environment an encrypted file of substantially-constant size representing log entries of selected events which occurred at the remote protected processing environment;

determining a decrypting key for the encrypted file and decrypting the log file;

analyzing the log entries of selected events at the remote processing environment and determining whether the operation of the protected processing environment is appropriate; and

reporting the results of the analyzing step.

16. A service providing the steps of Claim 15 and further including providing an instruction to initiate the logging of messages and an instruction to send to the log file to the remote location for analysis.

17. A service providing the steps of Claim 16 wherein the instruction to initiate logging of messages includes the step of initiating programming within the remote system to replace information in a log file with encrypted information relating to the operation of the remote system.

18. A service providing the steps of Claim 17 wherein the step of replacing data in the log file includes the step of replacing random data which was placed in the log file when it was created.

19. A service providing the steps of Claim 17 wherein the step of replacing data in the log file includes the step of using a pointer to the next location in the log file and the pointer wraps to the top of the log file after the log file has been filled.

20. Software stored on a device comprising:

a first module including stored program instructions for recording events

a second module for encrypting the recording of events using a key;

a third module for recording the encrypted events sequentially in a storage block of a substantially fixed size;

a fourth module maintaining a pointer of the next available location for the log;

a fifth module for responding to a command and sending the encrypted log file to a remote location for decryption and analysis.

21. Software including the elements of Claim 20 wherein the software further includes:

a mechanism for initializing the storage block of a fixed size with random information which has been encrypted to provide a block of apparent data.

22. Software including the elements of Claim 20 wherein the software further includes a module for writing the encrypted recorded events in a sequential order in the fixed-size storage and for wrapping around when the end of the fixed-size memory is reached.